***SNAKE MAZE GAME***

**INTRODUCTION**

A Snake Game is an **arcade maze game** which has been developed by Gremlin Industries and published by Sega in October 1976.

It is considered to be a skillful game and has popularized among people for generations. The snake in the Snake game is controlled using the four direction buttons relative to the direction it is headed in. The player’s objective in the game is to achieve maximum points as possible by collecting food or fruits. The player loses once the snake hits the wall or hits itself.

For the python beginners, those who are interested in making something easier in your domain can definitely try this out and the module **Turtle**was made exactly for this purpose for the beginners to try out and can also submit as a part of the project. This program will be done

in **Python 3**.

**SOURCE CODE**

**import turtle**

**import random**

**w = 500**

**h = 500**

**fs = 10**

**d = 100 # milliseconds**

**offsets = {**

**"up": (0, 20),**

**"down": (0, -20),**

**"left": (-20, 0),**

**"right": (20, 0)**

**}**

**def r():**

**global saap, kata, khanaT, pen**

**saap = [[0, 0], [0, 20], [0, 40], [0, 60], [0, 80]]**

**kata = "up"**

**khanaT = nun()**

**food.goto(khanaT)**

**hall()**

**def hall():**

**global kata**

**new\_head = saap[-1].copy()**

**new\_head[0] = saap[-1][0] + offsets[kata][0]**

**new\_head[1] = saap[-1][1] + offsets[kata][1]**

**if new\_head in saap[:-1]:**

**r()**

**else:**

**saap.append(new\_head)**

**if not khana():**

**saap.pop(0)**

**if saap[-1][0] > w / 2:**

**saap[-1][0] -= w**

**elif saap[-1][0] < - w / 2:**

**saap[-1][0] += w**

**elif saap[-1][1] > h / 2:**

**saap[-1][1] -= h**

**elif saap[-1][1] < -h / 2:**

**saap[-1][1] += h**

**pen.clearstamps()**

**#clears all the stamps**

**for segment in saap:**

**pen.goto(segment[0], segment[1])**

**pen.stamp()**

**screen.update()**

**#updates the turtle.screen screen**

**turtle.ontimer(hall, d)**

**def khana():**

**global khanaT**

**if dist(saap[-1], khanaT) < 20:**

**khanaT = nun()**

**food.goto(khanaT)**

**return True**

**return False**

**def nun():**

**x = random.randint(- w / 2 + fs, w / 2 - fs)**

**y = random.randint(- h / 2 + fs, h / 2 - fs)**

**return (x, y)**

**def dist(poos1, poos2):**

**x1, y1 = poos1**

**x2, y2 = poos2**

**distance = ((y2 - y1) \*\* 2 + (x2 - x1) \*\* 2) \*\* 0.5**

**return distance**

**def mathi():**

**global kata**

**if kata != "down":**

**kata = "up"**

**def go\_right():**

**global kata**

**if kata != "left":**

**kata = "right"**

**def go\_down():**

**global kata**

**if kata != "up":**

**kata = "down"**

**def go\_left():**

**global kata**

**if kata != "right":**

**kata = "left"**

**screen = turtle.Screen()**

**screen.setup(w, h)**

**screen.title("saap")**

**screen.bgcolor("green")**

**screen.setup(500, 500)**

**pen = turtle.Turtle("square")**

**pen.penup()**

**food = turtle.Turtle()**

**food.shape("circle")**

**food.color("white")**

**food.shapesize(fs / 20)**

**food.penup()**

**screen.listen()**

**screen.onkey(mathi, "Up")**

**screen.onkey(go\_right, "Right")**

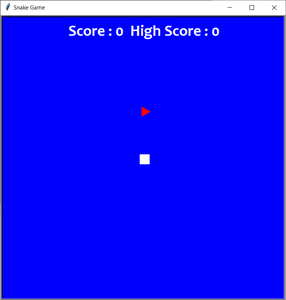
**screen.onkey(go\_down, "Down")**

**screen.onkey(go\_left, "Left")**

**r()**

**turtle.done()**

**CONCLUSION**



***Demonstrating the final game displaying everything written in code***

GLA UNIVERSITY

*Mini Project of Machine Learning*

*SNAKE MAZE GAME*